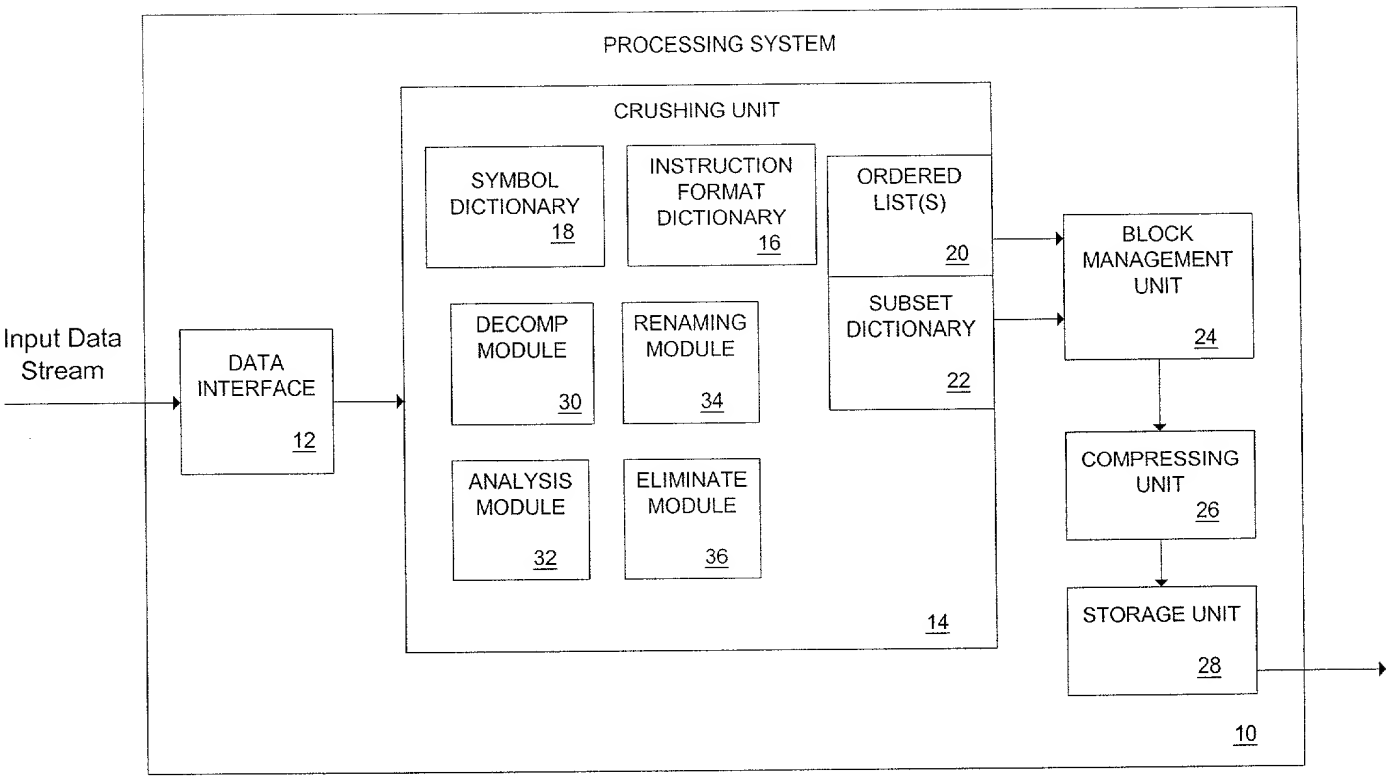


FIGURE 1A

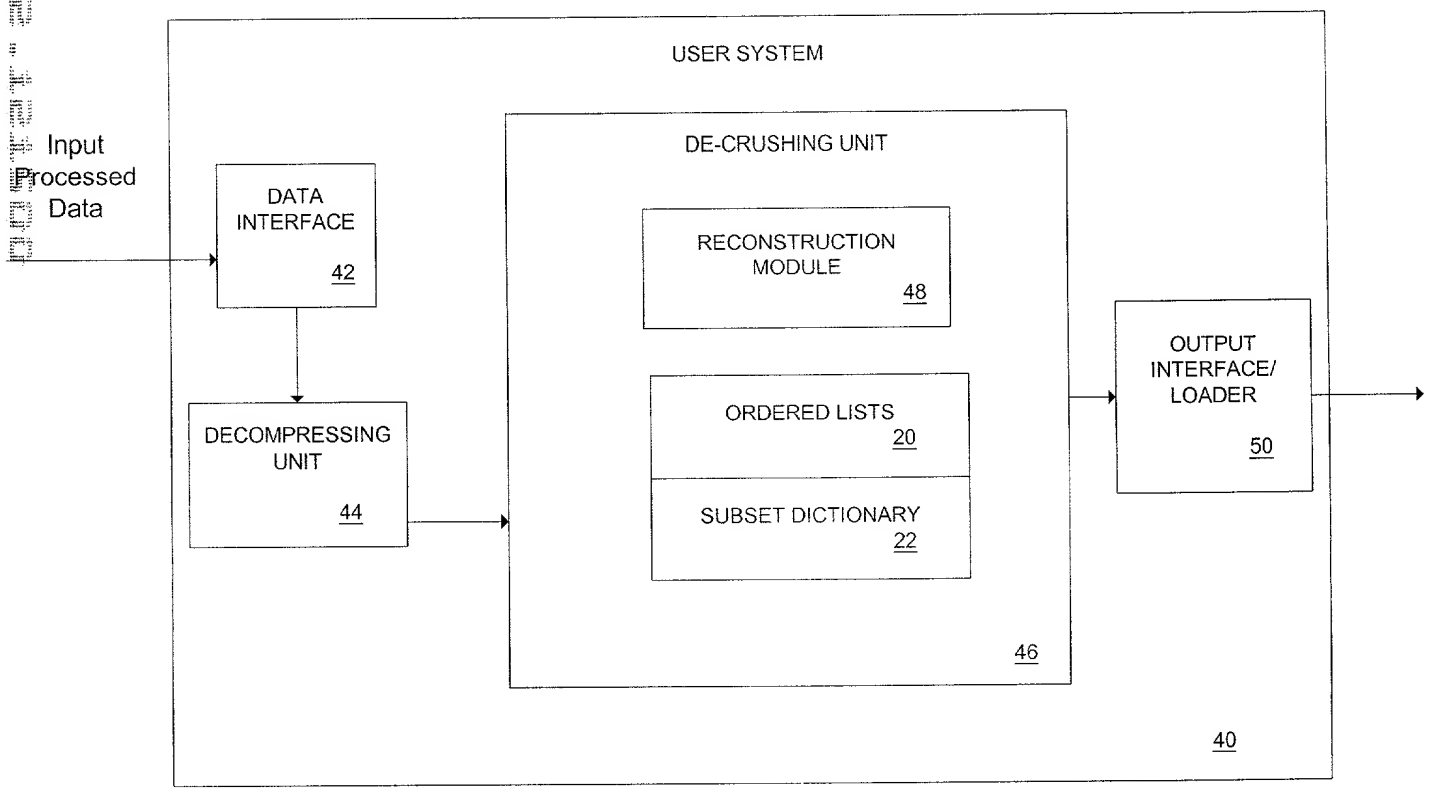


FIGURE 1B

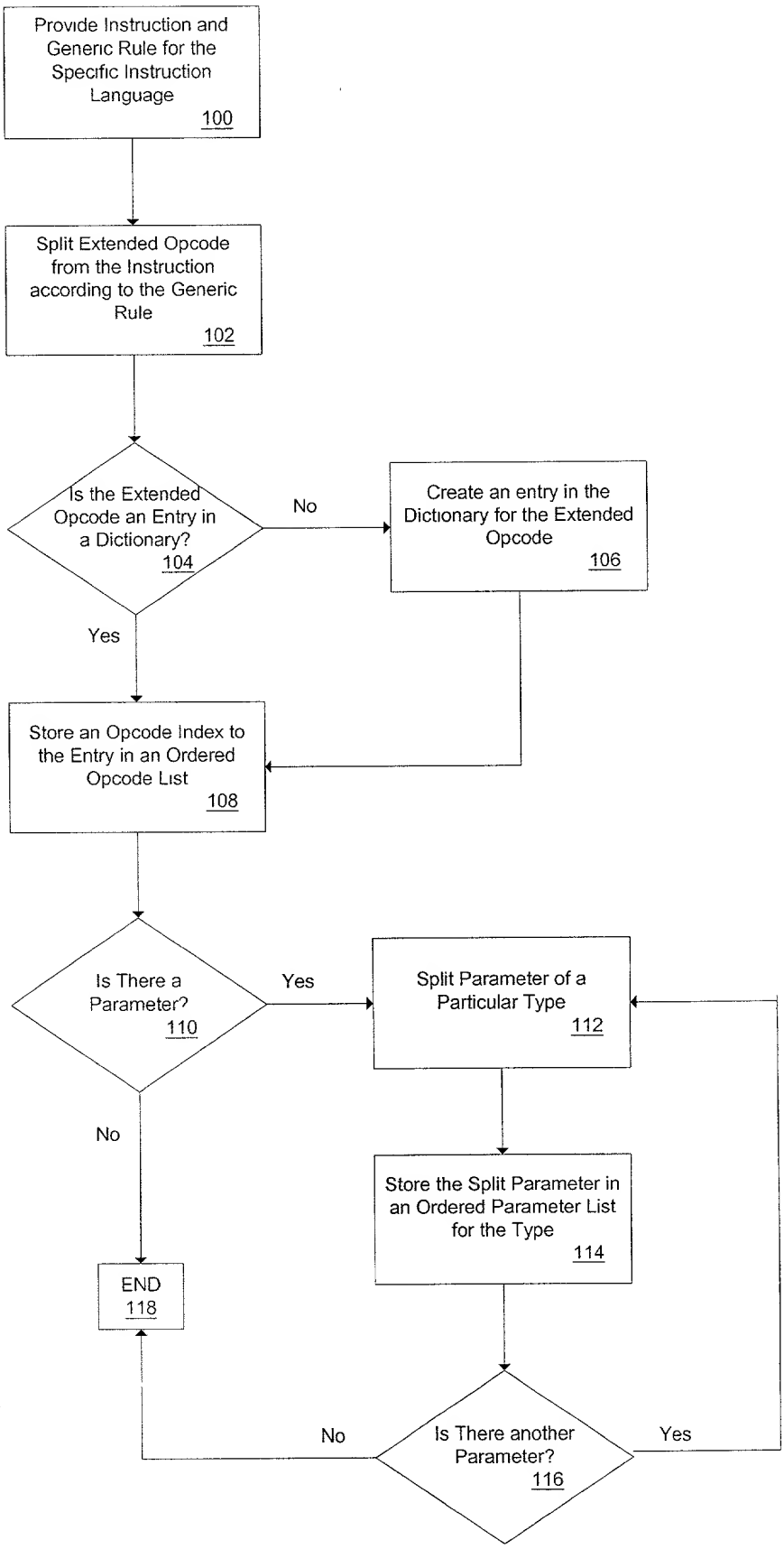

Provide Instruction and Generic Rule for the Specific Instruction Language
100
Split Extended Opcode from the Instruction according to the Generic Rule
102
Is the Extended Opcode an Entry in a Dictionary?
104
No
Create an entry in the Dictionary for the Extended Opcode
106
Yes
Store an Opcode Index to the Entry in an Ordered Opcode List
108
Is There a Parameter?
110
Yes
Split Parameter of a Particular Type
112
No
Store the Split Parameter in an Ordered Parameter List for the Type
114
END
118
Is There another Parameter?
116
No
Yes

Figure 2A

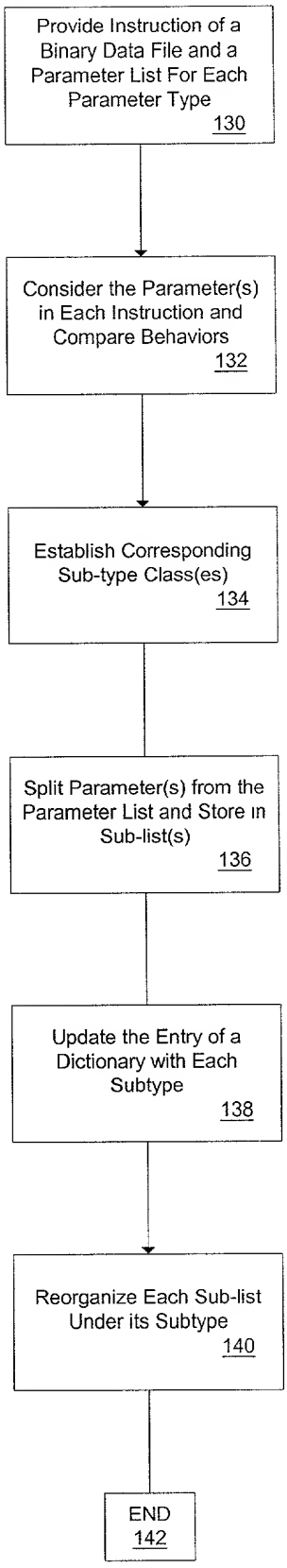

Provide Instruction of a Binary Data File and a Parameter List For Each Parameter Type
130
Consider the Parameter(s) in Each Instruction and Compare Behaviors
132
Establish Corresponding Sub-type Class(es)
134
Split Parameter(s) from the Parameter List and Store in Sub-list(s)
136
Update the Entry of a Dictionary with Each Subtype
138
Reorganize Each Sub-list Under its Subtype
140
END
142

Figure 2B

FIGURE 3A

16

52 Entry (Index)	56 Prefix	58 OpCode	60 ModR/M	62 Parameter Type	64 Analysis Data
1		01		Imm1, Offset 4, SIB	
2		02		Imm1, Offset 4, SIB	
3	0 x 64	02	0x6E	Imm1, Offset 2	
4	0 x 64, 0 x 66	02	0x6E	Imm1, Offset 2	
5		01	0x6E	Imm2, Offset 4	

64

FIGURE 3B

22

72 INDEX	74 EXTENDED OPCODE	76 PARAMETER LIST INDEX
1	01	1, 7, 9
2	02	1, 7, 9
3	64, 02, 6E	1, 6
4	64, 66, 02, 6E	1, 6
5	01, 6E	2, 7

FIGURE 3C

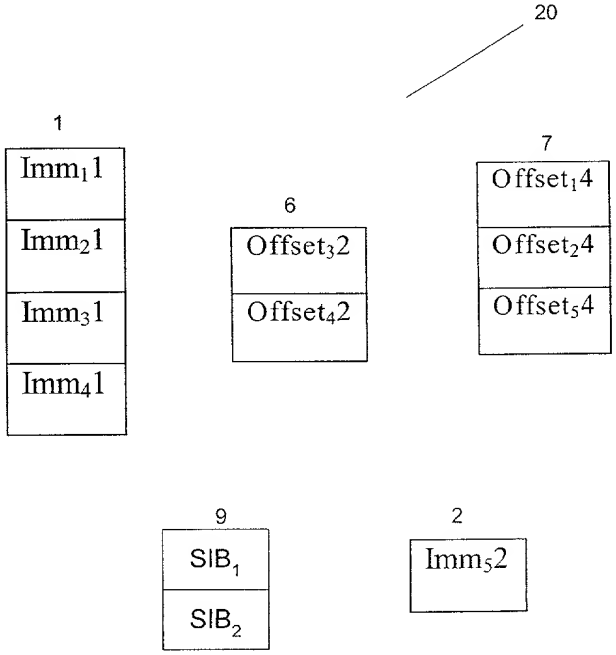
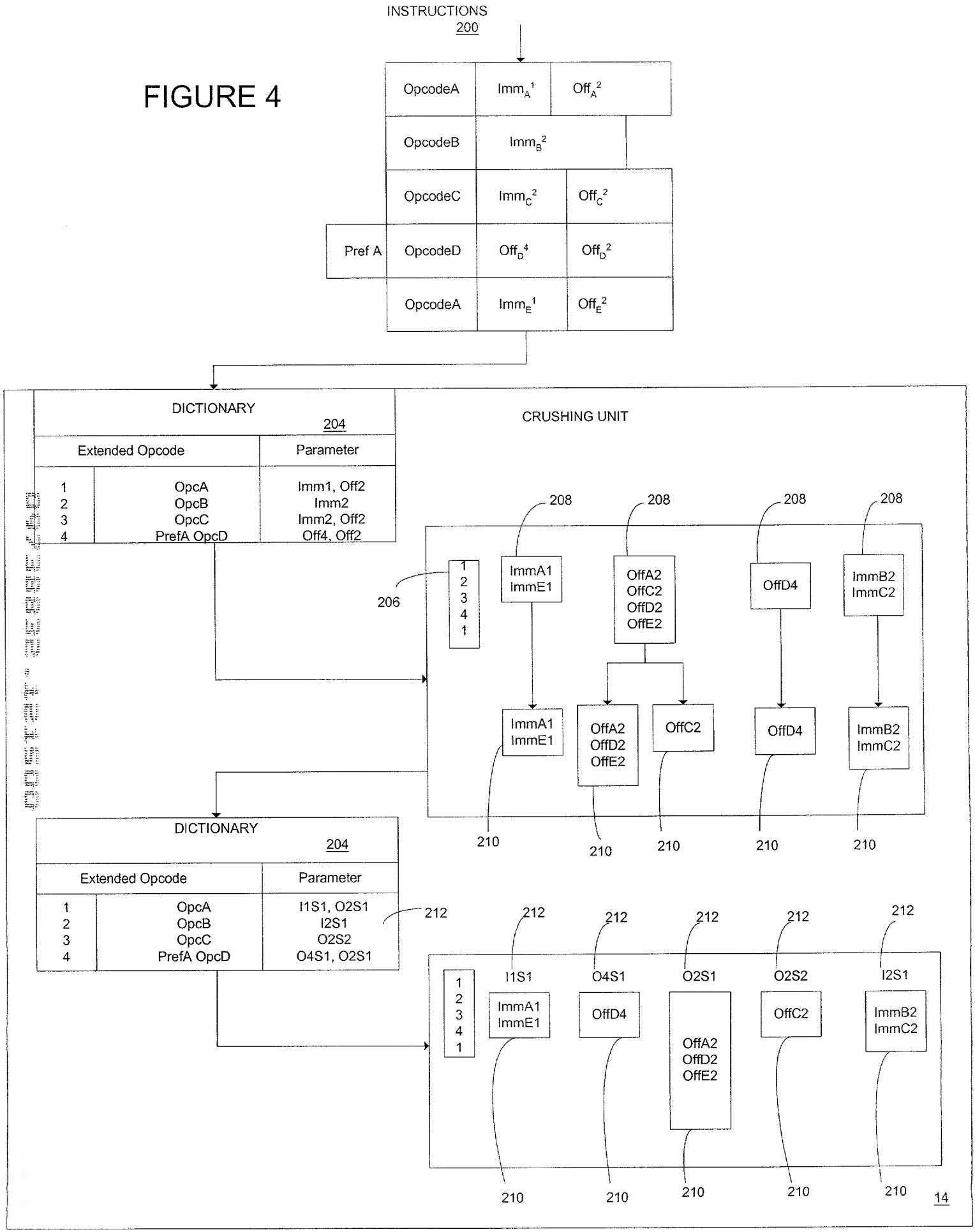


FIGURE 4


INSTRUCTIONS
200
OpcodeA
Imm_A^1
Off_A^2
OpcodeB
Imm_B^2
OpcodeC
Imm_C^2
Off_C^2
Pref A
OpcodeD
Off_D^4
Off_D^2
OpcodeA
Imm_E^1
Off_E^2
DICTIONARY
204
Extended Opcode
Parameter
1 OpcA Imm1, Off2
2 OpcB Imm2
3 OpcC Imm2, Off2
4 PrefA OpcD Off4, Off2
CRUSHING UNIT
208
206
1 2 3 4 1
ImmA1 ImmE1
OffA2 OffC2 OffD2 OffE2
OffD4
ImmB2 ImmC2
210
DICTIONARY
204
Extended Opcode
Parameter
1 OpcA I1S1, O2S1
2 OpcB I2S1
3 OpcC O2S2
4 PrefA OpcD O4S1, O2S1
212
I1S1
O4S1
O2S1
O2S2
I2S1
ImmA1 ImmE1
OffD4
OffA2 OffD2 OffE2
OffC2
ImmB2 ImmC2
14

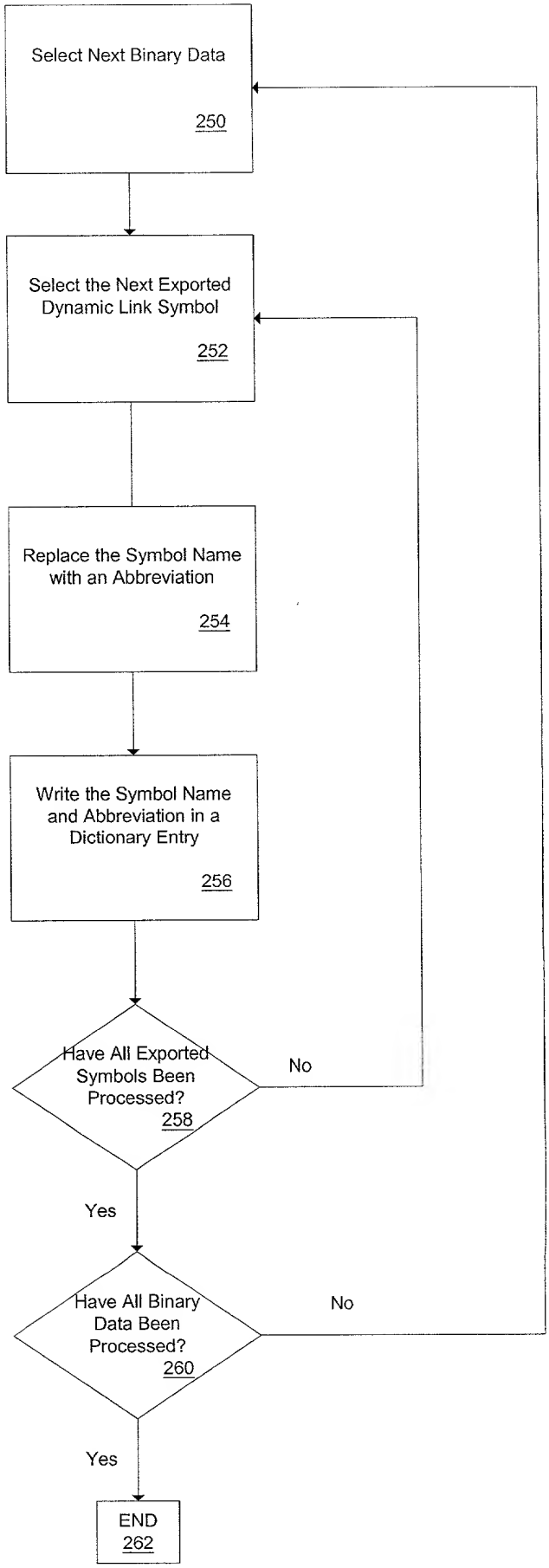

Select Next Binary Data
250
Select the Next Exported Dynamic Link Symbol
252
Replace the Symbol Name with an Abbreviation
254
Write the Symbol Name and Abbreviation in a Dictionary Entry
256
Have All Exported Symbols Been Processed?
258
No
Yes
Have All Binary Data Been Processed?
260
No
Yes
END
262

Figure 5A

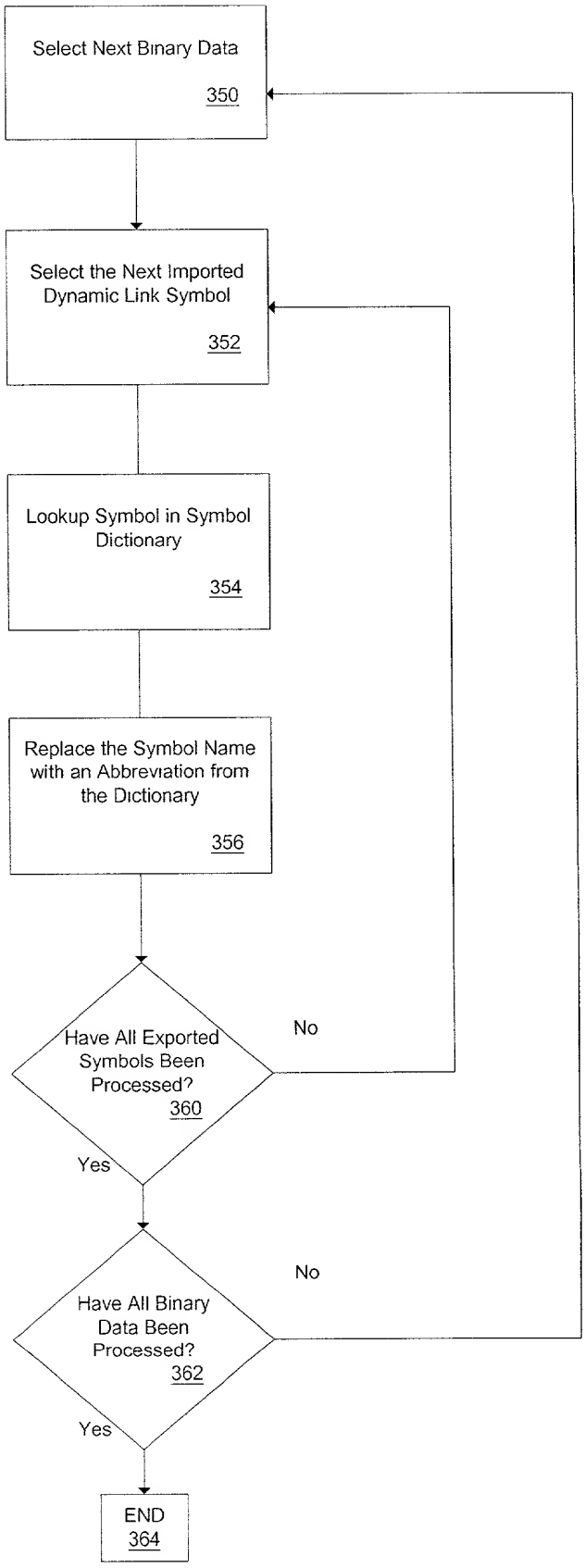
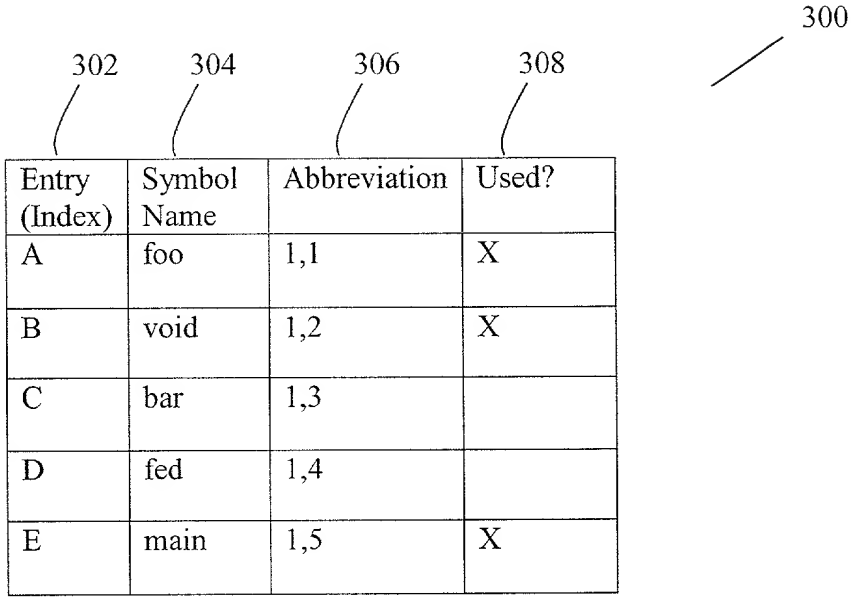

Select Next Binary Data
350
Select the Next Imported Dynamic Link Symbol
352
Lookup Symbol in Symbol Dictionary
354
Replace the Symbol Name with an Abbreviation from the Dictionary
356
Have All Exported Symbols Been Processed?
360
No
Yes
Have All Binary Data Been Processed?
362
No
Yes
END
364

Figure 5B

FIGURE 6



Entry (Index)	Symbol Name	Abbreviation	Used?
A	foo	1,1	X
B	void	1,2	X
C	bar	1,3	
D	fed	1,4	
E	main	1,5	X

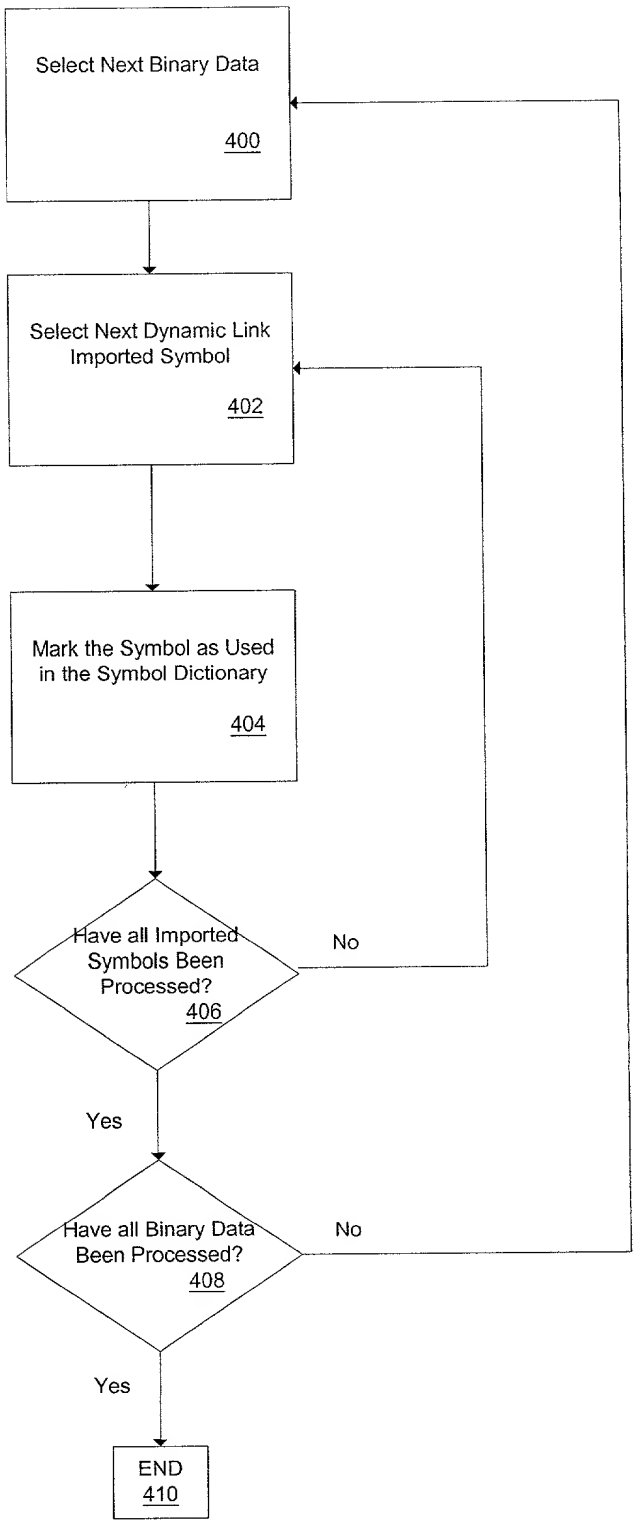

Select Next Binary Data
400
Select Next Dynamic Link Imported Symbol
402
Mark the Symbol as Used in the Symbol Dictionary
404
Have all Imported Symbols Been Processed?
406
No
Yes
Have all Binary Data Been Processed?
408
No
Yes
END
410

Figure 7A

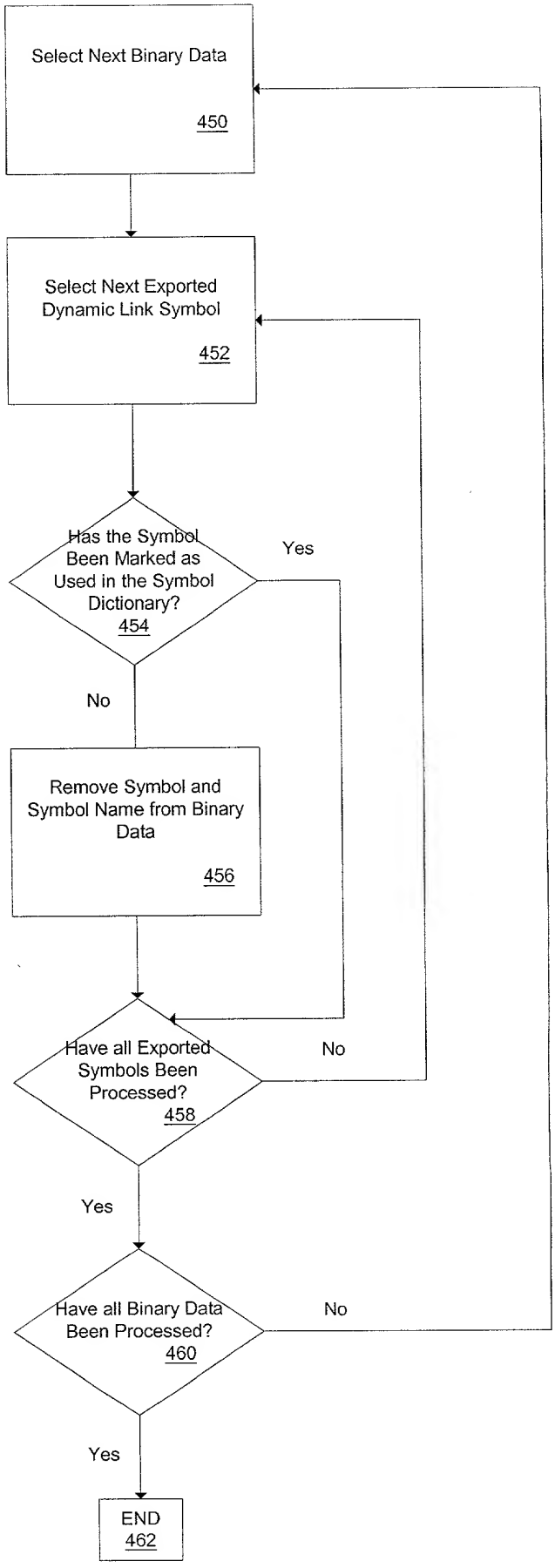

Select Next Binary Data
450
Select Next Exported Dynamic Link Symbol
452
Has the Symbol Been Marked as Used in the Symbol Dictionary?
454
Yes
No
Remove Symbol and Symbol Name from Binary Data
456
Have all Exported Symbols Been Processed?
458
No
Yes
Have all Binary Data Been Processed?
460
No
Yes
END
462

Figure 7B

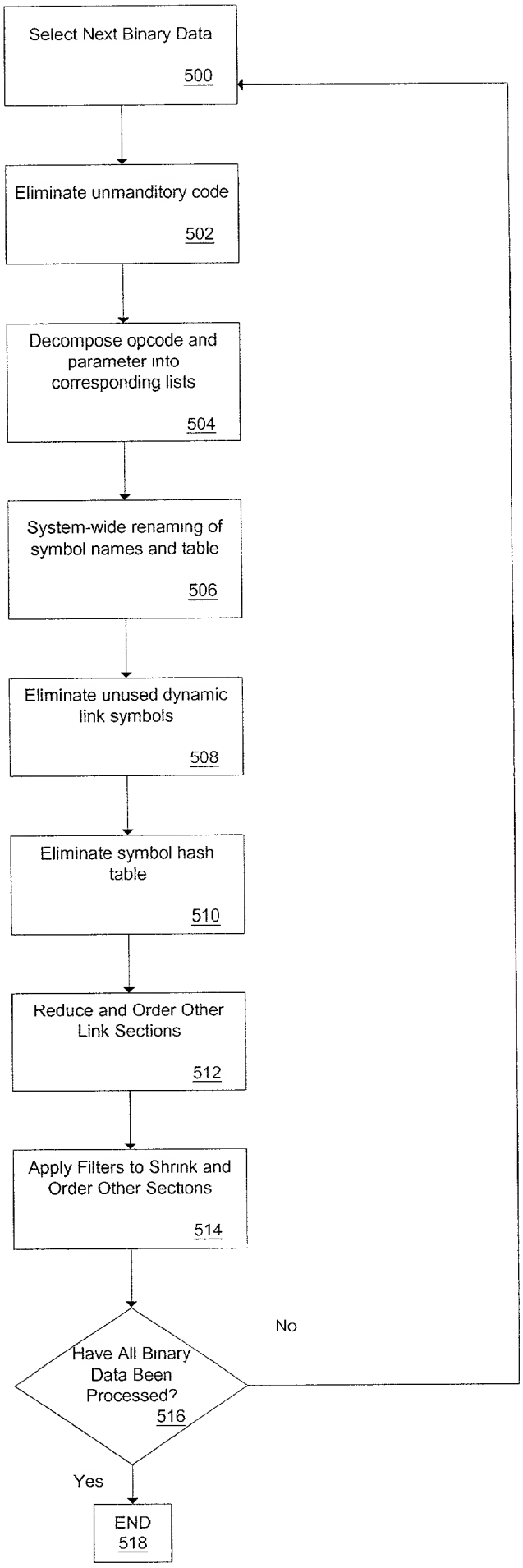

Select Next Binary Data
500
Eliminate unmanditory code
502
Decompose opcode and parameter into corresponding lists
504
System-wide renaming of symbol names and table
506
Eliminate unused dynamic link symbols
508
Eliminate symbol hash table
510
Reduce and Order Other Link Sections
512
Apply Filters to Shrink and Order Other Sections
514
Have All Binary Data Been Processed?
516
No
Yes
END
518

Figure 8

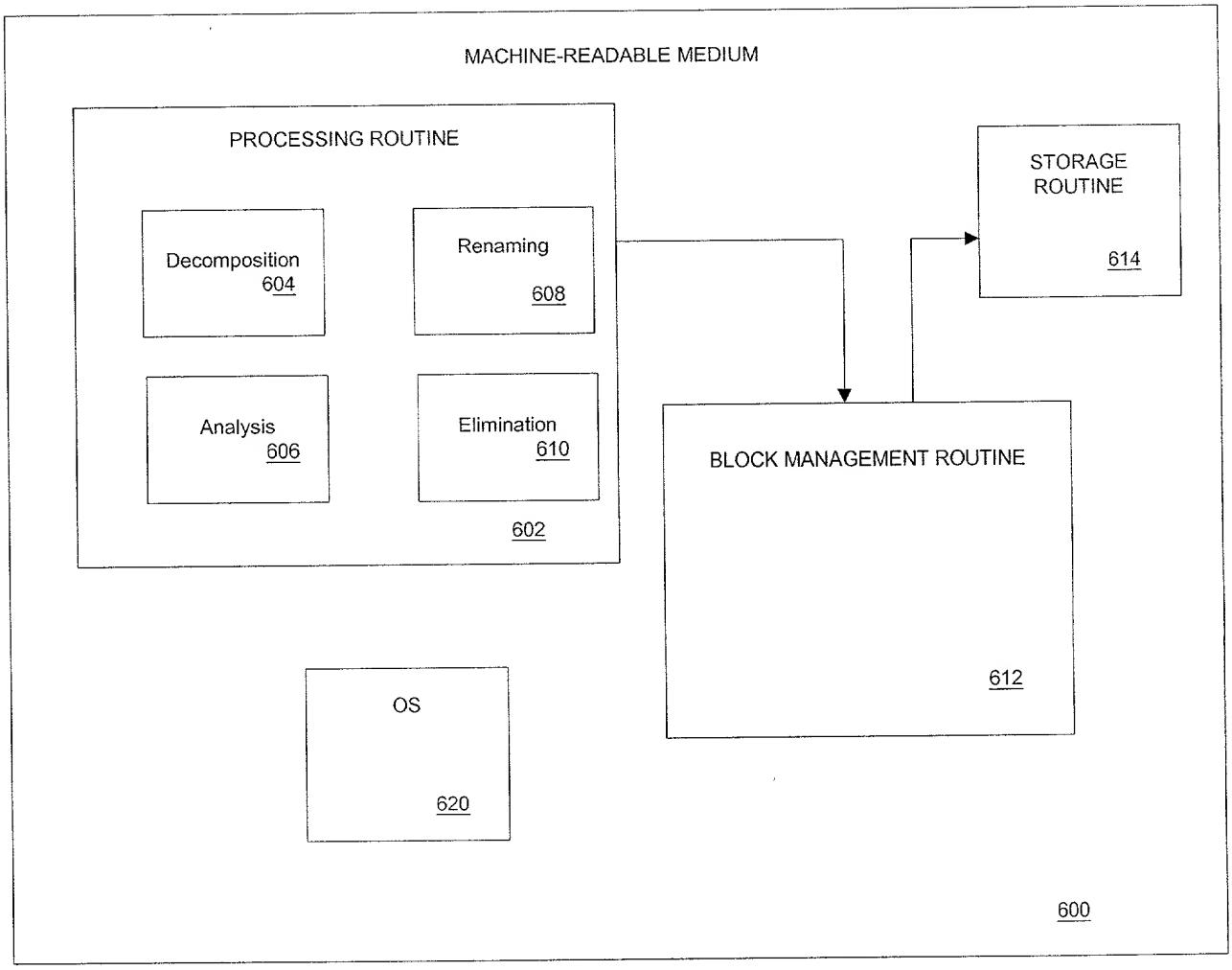

MACHINE-READABLE MEDIUM
PROCESSING ROUTINE
Decomposition
604
Renaming
608
Analysis
606
Elimination
610
602
STORAGE
ROUTINE
614
BLOCK MANAGEMENT ROUTINE
612
OS
620
600

FIGURE 9